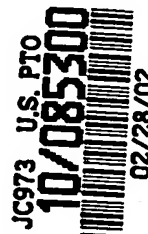


02/28/02



ASSISTANT COMMISSIONER OR INTS
WASHINGTON, D.C. 20231

DC ET NUMBER: GB920010075US1



Transmitted herewith for filing is the Patent Application of:

ASSIGNEE NAME: International Business Machines Corporation
ASSIGNEE RESIDENCE: Armonk, New York

Inventor: S A J Holdsworth, P R MacFarlane

For: METHOD AND SYSTEM FOR PRESERVING MESSAGE ORDER WHEN PARALLEL PROCESSING MESSAGES

Enclosed are:

- ☒ Patent Specification and Declaration
- ☒ 2 sheets of drawing(s)
- ☒ An assignment of the invention to International Business Machines Corporation (includes Recordation Form Cover Sheet)
- ☒ A certified copy of a United Kingdom application filed on 12 December 2001, Serial Number 0129672.2
- ☒ Information Disclosure Statement, PTO 1449 and copies of references

The fee has been calculated as shown below:

	Number Filed	Number Extra	Rate	Fee \$
Basic Fee				\$740.00
Total Claims	20-20	x	18	\$0.00
Independent Claims	2-3	x	84	\$0.00
MULTIPLE DEPENDENT CLAIM PRESENTED			280	\$0.00
			TOTAL	\$740.00

- ☒ Please charge my Deposit Account No. 09-0461 in the amount of \$740.00. A duplicate of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account 09-0461. A duplicate copy of this sheet is enclosed.
- ☒ Any additional filing fees required under 37 CFR §1.16.
- ☒ Any patent application processing fees under 37 CFR §1.17.

EXPRESS MAIL CERTIFICATE

Express Label No ET944325950US1

Date of Deposit February 28, 2002

I hereby certify that this paper and fee are being deposited with the United States Postal Service Express Mail Post Office to Addressee service under 37 CFR § 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231. The Applicant and/or Attorney requests the date of deposit as the filing date.

Person mailing paper/fee Dianne Lane

Signature

Dianne Lane

Respectfully submitted

Jeanine S. Ray-Yarlett's
Registration No. 39,808
IBM Corp, IP Law Dept T81/503,
3039 Cornwallis Road,
PO Box 12195,
Research Triangle Park, NC 27709-2195.
Telephone: (919)543 -2541
Fax: (919) 254-4330

Customer Number: 25259



INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ



I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

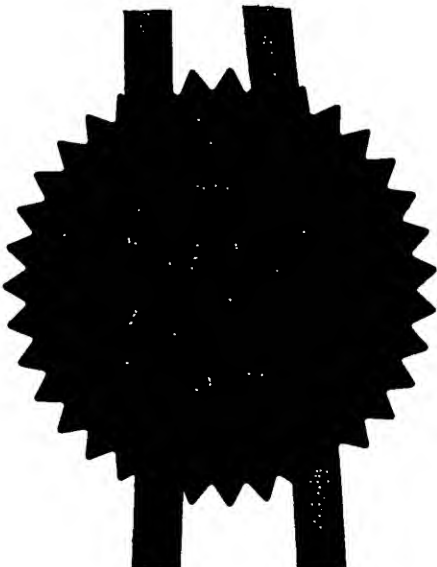
In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

CERTIFIED COPY OF PRIORITY DOCUMENT

Signed

Dated 4 February 2002



THE PATENT OFFICE
F
12 DEC 2001
NEWPORT

The
Patent
Office

1/77

Patents Act 1977
Rule 16

12DEC01 E681275-1 000611
P01/7700 0.00-0129672.2

Request for grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920010075GB1		
2.	Patent application number (The Patent Office will fill in this part)	0129672.2		12 DEC 2001
3.	Full name, address and postcode of the or of each applicant (underline all surnames)	INTERNATIONAL BUSINESS MACHINES CORPORATION Armonk New York 10504 United States of America		
	Patents ADP number (if you know it)	519631001		
	If the applicant is a corporate body, give the country/state of its incorporation	State of New York United States of America		
4.	Title of the invention	METHOD AND SYSTEM FOR PRESERVING MESSAGE ORDER WHEN PARALLEL PROCESSING MESSAGES		
5.	Name of your agent (if you have one)	M J Jennings		
	"Address for Service" in the United Kingdom to which all correspondence should be sent (including the postcode)	IBM United Kingdom Limited Intellectual Property Department Hursley Park Winchester Hampshire SO21 2JN		
	Patents ADP number (if you know it)	919005		
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority App No (if you know it)	Date of filing (day/month/year)
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date or the earlier application	No of earlier application	Date of filing (day/month/year)	

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if:
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.) Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description 6

Claim(s) 3

Abstract 1

Drawing(s) 2 x 2 8w

10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right to grant of a patent (Patents Form 7/77) 3

Request for preliminary examination and search (Patents Form 9/77)

Request for substantive examination (Patents Form 10/77)

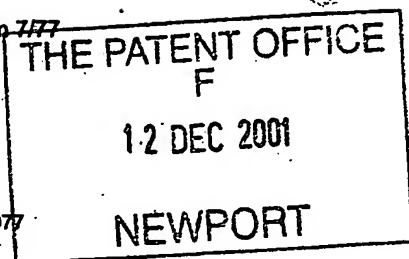
Any other documents (please specify)

11. I/We request the grant of a patent on the basis of this application


Signature

11 December
2001
Date

12. Name and daytime telephone number of person to contact in the United Kingdom M J Jennings
01962 816725



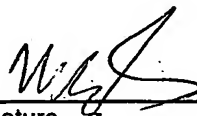
The
Patent
Office

7/77

Patents Act 1977
Rule 15

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920010075GB1
2.	Patent application number (if you know it)	0129672.2 12 DEC 2001
3.	Full name of the or of each applicant	INTERNATIONAL BUSINESS MACHINES CORPORATION
4.	Title of invention	METHOD AND SYSTEM FOR PRESERVING MESSAGE ORDER WHEN PARALLEL PROCESSING MESSAGES
5.	State how the applicant(s) derived the right from the inventor(s) to be granted a patent	By employment and agreement
6.	How many, if any, additional Patents Forms 7/77 are attached to this form?	
7.	I/We believe that the person(s) named over the page (and on any extra copies of this form) is/are the inventor(s) of the invention which the above patent application relates to.	
	Signature 	11 December 2001 Date
8.	Name and daytime telephone number of person to contact in the United Kingdom	M J Jennings Tel: 01962 816725

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Simon Antony James HOLDSWORTH
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number (if known)

7127608002

Peter Robert MacFARLANE
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number (if known)

8287195001

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

REMINDER

Have you signed the form?

Patents ADP number (if known)

METHOD AND SYSTEM FOR PRESERVING MESSAGE ORDER
WHEN PARALLEL PROCESSING MESSAGES

5 Field of the Invention

This invention relates to processing of messages in a computer system.

10 Background of the Invention

In the field of this invention it is known that for optimum performance of message throughput when processing messages in a computer system, there is a requirement to take advantage (if possible) of parallel processing facilities of the system on which the processing takes place, for example employing multiple parallel threads in a process where each thread processes a separate message.

15 It is known that an input node of a computer program or data processing system which is receiving messages may be configured to process messages containing the same userid in sequence. In this case messages containing different userids may overtake each other, but messages with the same userid will not. Such message processing works acceptably where the message is processed by a single message "broker" (which provides routing/formatting services) and is then sent to a client application.

20 However, in the situation where the message is passed from one broker to another in a broker network before it reaches a client, this approach is not suitable because all messages arriving at an input node of a second broker have the same userid (the userid of the previous broker to process a message), and the same is true at subsequent brokers, so the userid is not an adequate parameter for preserving message ordering.

30 A need therefore exists to preserve message order when parallel processing messages wherein the abovementioned disadvantage may be alleviated.

35 Statement of Invention

40 In accordance with a first aspect of the present invention there is provided a method for preserving message order when parallel processing messages as claimed in claim 1.

In accordance with a second aspect of the present invention there is provided a system for preserving message order when parallel processing messages as claimed in claim 8.

Brief Description of the Drawings

One system and method, for preserving message order when parallel processing messages, incorporating the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 shows a block-schematic diagram of computer system incorporating the present invention; and

FIG. 2 shows a schematic representation of a message used in the computer system of FIG. 1.

Description of Preferred Embodiment

The basis of this invention, in a preferred embodiment, is that messages for which it is required to preserve the order of processing hold a marker to indicate that ordering is required. Further, the marker indicates the unique source of ordered messages such that, ~~the~~ messages from different sources are processed by a single message processing program (such as within a message brokering program), it is possible for the processor of the messages to determine whether processing of each message may be undertaken in parallel with messages already being processed without the possibility of the order of ordered messages being lost.

Referring now to FIG. 1, a computer system has a series of nodes N1, N2, N3 ... Nn, each of which produces a sequence of messages M1, M2, M3 ... Mm (the contents of which will be described in more detail below), resulting in a possible stream of messages (first to last):
N1M1, N2M1, N2M2, N2M3, N3M1, N4M1, N1M2, N2M4, N3M2, N4M2, N4M3, N2M5.

The resulting message stream is passed to a dispatcher D, which (as will be explained in greater detail below) dispatches selected messages in the message stream to different ones of parallel processing elements P1, P2 and P3, which together form a processor P. It will be understood that although the processor P is represented as a number of discrete parallel processors P1, P2 and P3, in practice the parallel processors need not be separate hardware processors and may be provided by respective threads of multi-threading software on a single hardware processor system. In the context of the present application, a 'parallel processor' should be

interpreted as optionally comprising a thread of a multi-threading computer program.

5 The system of FIG. 1 finds particular application in a message delivery system (not shown) in which messages may be delivered through a network of servers including one or more "brokers" which provide routing and formatting services. The brokers are located at communication hubs within the network. In such a network the nodes N1, N2, N3 ... Nn, may be
10 respective input nodes of a message broker in a sequence of message brokers, the input nodes N1, N2, N3 ... Nn, receiving messages processed by brokers (not shown) earlier in the sequence.

15 IBM Corporation's MQSeries and WebSphere MQ family of messaging products are examples of known products which support interoperation between application programs running on different systems in a distributed heterogeneous environment. Message queuing and commercially available message queuing products are described in "Messaging and Queuing Using the MQI", B.Blakeley, H.Harris & R.Lewis, McGraw-Hill, 1994, and in the
20 following publications which are available from IBM Corporation: "An Introduction to Messaging and Queuing" (IBM Document number GC33-0805-00) and "MQSeries - Message Queue Interface Technical Reference" (IBM Document number SC33-0850-01). The network via which the computers communicate using message queuing may be the Internet, an intranet, or any computer network. IBM, WebSphere and MQSeries are trademarks of IBM Corporation.

25 The message queuing inter-program communication support provided by the MQSeries products enables each application program to send messages to the input queue of any other target application program and each target application can asynchronously take these messages from its input queue for
30 processing. This achieves delivery of messages between application programs which may be spread across a distributed heterogeneous computer network, without requiring a dedicated logical end-to-end connection between the application programs, but there can be great complexity in the map of possible interconnections between the application programs.

35 This complexity can be greatly simplified by including within the network architecture a communications hub to which other systems connect, instead of having direct connections between all systems. Message brokering capabilities can then be provided at the communications hub to provide
40 intelligent message routing and integration of applications. Message brokering functions typically include the ability to route messages intelligently according to business rules and knowledge of different application programs' information requirements, using message 'topic' information contained in message headers, and the ability to transform

message formats using knowledge of the message format requirements of target applications or systems to reconcile differences between systems and applications.

Such brokering capabilities are provided, for example, by IBM Corporation's MQSeries Integrator (MQSI) and WebSphere MQ Integrator products, providing intelligent routing and transformation services for messages which are exchanged between application programs using IBM's MQSeries and WebSphere MQ messaging products.

The system of FIG. 1 can be used, for example, in an implementation where the processing nodes are MQSI Brokers, and the inter-broker communication employs MQ message queues, where the input node of each flow within the broker has an ordered attribute. If multiple input nodes or multiple flows produce messages that are merged into a single flow of messages, and some of these originate from nodes that require ordering to be preserved, the flow of messages may be sent between adjacent brokers.

Now, hypothetically for this example, if messages produced at odd nodes (N1, N3 etc) are to be treated as ordered, and at even nodes (N2, N4 etc) treated as unordered, then the example stream may be received and processed taking advantage of parallel processing for each message up to receipt of N1M2. When this message is encountered it must not be processed in parallel with N1M1, or ordering may be lost. Thus new threads may be dispatched as each message is received until a message is encountered which must preserve relative ordering with a message already being processed. When this condition occurs, processing of the first message must complete before the second can begin.

Referring now to FIG. 2, in an example of inter-broker message flows, each message M would contain a marker H (typically within a message header; which is combined with message content C) which is used to identify each unique source of messages for which it is required to preserve the message order. The marker would correspond to (or be derived from) a zero value if no ordering of the message is necessary, and a non-zero value if ordering is required. The non-zero value would be derived in such a way that it would be likely to be unique amongst a group of message sources.

The dispatcher D, which initiates parallel processing of the messages, retains a list of all markers of messages that are being processed in parallel. When a message becomes available to the dispatcher D for processing, the dispatcher inspects the list of messages that are currently being processed to determine whether the marker of this message is in the list. If the marker is found in the list, the dispatcher D does

not initiate parallel processing for this message until the marker is no longer in the list.

In this way, the dispatching algorithm used by the receiving inter-broker node would simply dispatch new threads, up to a configured limit, until a message was received that contained a marker that matched the marker of a message currently being processed. At this time the dispatcher D would wait for processing of this message to complete before processing further messages.

Referring to FIG. 2, the marker H used in the ordered message may be constructed as a hash-code derived from unique characteristics of the source of the ordered messages where these characteristics may be unique to the sequence of ordered messages. For example, characteristics may include a combination of the originating userid of the messages, the queue on which messages are put; an identifier associated with the input node; and an identifier associated with the mode of processing, etc. If two (or more) sources of messages happened to generate the same marker, the ordering of messages from the sources would still be preserved - the only adverse effect would be that unnecessary serialization of messages would occur when parallel processing theoretically would be possible.

For example, the marker H may be a hash value formed from the message's originating userid and a zero/non-zero value as follows:

At input node N1, operating in a mode that requires processing of all messages sequentially (using a single thread)

message N1M1: userid 'fred', value 234325;

message N1M2: userid 'bill', value 534345;

message N1M3: userid 'fred', value 234325.

At input node N2, operating in a mode that requires processing of messages on multiple threads

message N2M1: userid 'bill', value 0;

message N2M2: userid 'fred', value 0.

In the example given above, when the messages N1M1, N1M2, N1M3, N2M1 & N2M2 arrive at the dispatcher D, the dispatcher can tell that it can process messages N1M1, N1M2, N2M1 & N2M2 in parallel, but it must wait until message N1M1 has finished processing before message N1M3 can be processed.

It will be appreciated that the hash value is a compact representation of the data which only needs a simple comparison to

determine whether serialisation is required. It will also be appreciated that the userid per se of the message originator does not need to be disclosed to other brokers or subscribers; the hash value is sent, analysed on receipt and used to ensure serialisation as necessary of a subset of messages.

It will be understood that by delaying initiating parallel processing of the new message until the marker is no longer in the list, as described above, the invention may delay initiating parallel processing of any message awaiting dispatch until the marker of the next message on the input stream is no longer in the list. Clearly, this could be inefficient.

A possible enhancement to avoid this inefficiency would be for the dispatcher to maintain an ordered queue for each marker that is in its list of messages being processed. When message processing by a parallel processing element completes for a specific marker, the dispatcher then assigns to this processing element the next message in the ordered queue for this specific marker. If the queue for a specific marker is empty, the dispatcher then selects the next message in its input stream of messages awaiting dispatch: if this message has a marker in the list then the message is added to the ordered queue for this marker, if not then the message is assigned to a processing element and its marker added to the list.

The advantage of this enhancement would be that it enables the parallel processing capacity to always be fully exploited when further unordered (or first in a new order) messages are available to process.

It will be appreciated that the method described above for preserving message order when parallel processing messages in a computer system will typically be carried out by software running on a data processing system (not shown), and that the software may be provided as a computer program element carried on any suitable data carrier (also not shown) such as a machine-readable magnetic or optical recording medium.

It will be understood that the method and system described above allows a stream of messages originating from multiple sources to be processed with the advantages of parallel processing, while preserving the order of a subset of these messages.

an originating userid of the message;

an identifier associated with a respective input node receiving the message; and

5

an identifier associated with the mode of processing.

8. The method of any one of claims 5 to 7, wherein said characteristics include:

10

an originating userid of the message;

an identifier of a repository on which the message is put; and

15

an identifier associated with the mode of processing.

9. The method of any preceding claim wherein the marker comprises a hash value.

20

10. A system for preserving message order when parallel processing messages, comprising:

means, responsive to a marker within a message, for identifying a source of messages and determining whether it is required to preserve the message order; and

25

a dispatcher for dispatching each message in accordance with its marker to one of a plurality of parallel processors such that processing order is preserved when required for messages processed through the plurality of parallel processors.

30

11. The system of claim 10 wherein the dispatcher comprises:

a list of all markers of messages that are being processed in parallel;

35

means for determining whether the marker of a new message is present in the list; and

40

means for delaying initiating parallel processing of the new message until the marker is no longer in the list.

12. The system of claim 11 further comprising an ordered queue for each marker that is in the list of messages being processed, and the dispatcher

comprises means for, when message processing by one of the plurality of parallel processors completes for a marker, dispatching to said one of the plurality of parallel processors the next message in the ordered queue for said marker.

13. The system of claim 10, 11 or 12 wherein a predetermined value of the marker indicates that ordering is not required.

14. The system of any one of claims 10 to 13 wherein the marker is derived from characteristics of the source of the ordered messages.

15. The system of claim 14, wherein the characteristics include:

an originating userid of the message;

an identifier associated with a respective input node receiving the message; and

an identifier associated with the mode of processing.

16. The system of claim 14 or claim 15, wherein the characteristics include:

an originating userid of the message;

an identifier of a repository on which the message is put; and

an identifier associated with the mode of processing.

17. The system of any one of claims 10 to 16 wherein the marker comprises a hash value.

18. A computer program comprising computer program code for controlling the performance of a data processing apparatus on which it executes to perform a method according to any one of claims 1-9.

19. A method for preserving message order when parallel processing messages substantially as hereinbefore described with reference to any one of the accompanying drawings.

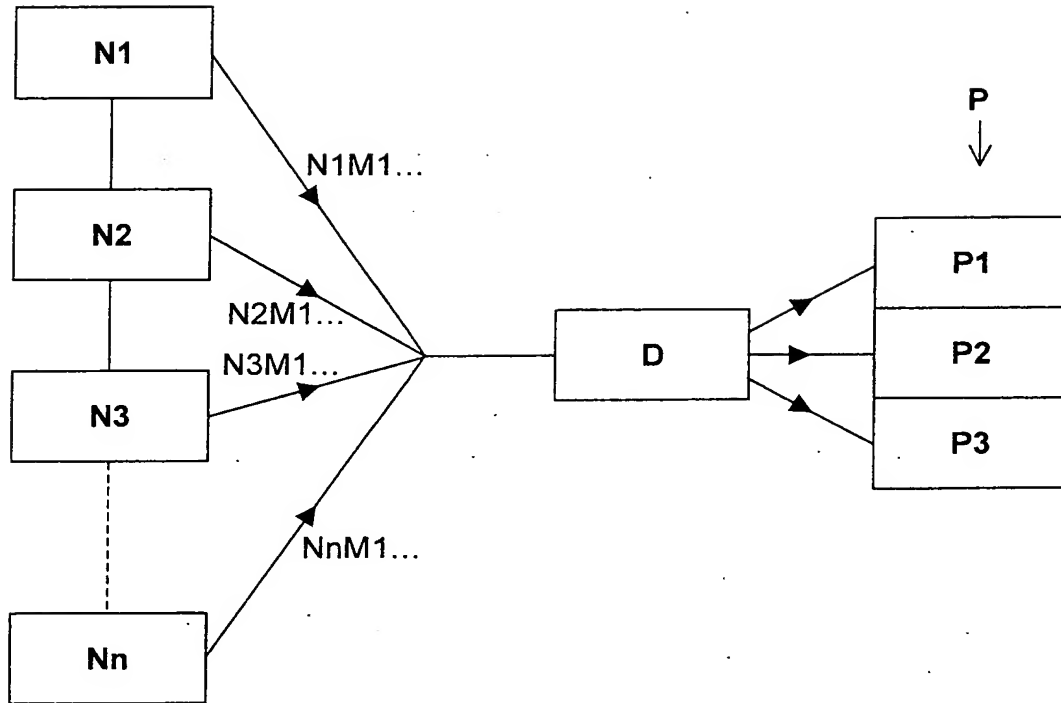
20. A system for preserving message order when parallel processing messages substantially as hereinbefore described with reference to any one of the accompanying drawings.

ABSTRACT

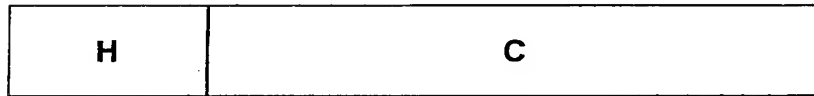
METHOD AND SYSTEM FOR PRESERVING MESSAGE ORDER
WHEN PARALLEL PROCESSING MESSAGES

5
10
15
20

A method and system for preserving message order when parallel processing messages by: providing a marker (H) in each message (M) for identifying a source of messages for which it is required to preserve the message order; and dispatching (D) each message in accordance with its marker to one of a plurality of parallel processors (P1-P3) so that processing is preserved for messages processed through the plurality of parallel processors. The dispatcher (D) retains a list of all markers of messages that are being processed in parallel, and when a message becomes available to the dispatcher for processing, the dispatcher inspects the list of messages that are currently being processed to determine whether the marker of this message is in the list, and if so delays initiating parallel processing for this message until the marker is no longer in the list. This technique allows a stream of messages originating from multiple sources to be processed with the advantages of parallel processing, while preserving the order of a subset of these messages.

**FIG. 1**

2/2



M ↗

FIG. 2